G06F 15/80

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 539 595 A1**

(12) EUROPEAN PATENT APPLICATION published in accordance with Art. 158(3) EPC

(21) Application number: 92908498.6

(22) Date of filing: 09.04.92

(86) International application number: PCT/JP92/00447

(87) International publication number: WO 92/18935 (29.10.92 92/27)

(51) Int. Cl.[5]: G06F ~~15/16, G06F 9/38~~

(30) Priority: 09.04.91 JP 76574/91

(43) Date of publication of application: 05.05.93 Bulletin 93/18

(84) Designated Contracting States: DE FR GB

(71) Applicant: FUJITSU LIMITED
1015, Kamikodanaka Nakahara-ku
Kawasaki-shi Kanagawa 211(JP)

(72) Inventor: KATSUNO, Akira
Fujitsu Limited 1015, Kamikodanaka
Nakahara-ku Kawasaki-shi Kanagawa
211(JP)

(74) Representative: Fane, Christopher Robin King et al
HASELTINE LAKE & CO. Hazlitt House 28
Southampton Buildings Chancery Lane
London, WC2A 1AT (GB)

(54) DATA PROCESSOR AND DATA PROCESSING METHOD.

(57) In a parallel data processing system in which processor elements (PE) are arranged in a two-dimensional grid form, each PE includes 1-bit arithmetic means for 1-bit operand data, storage means for storing operand data and/or the result, and communication means for effecting communication with other PEs. A common bus for connecting PEs in a transverse (row) direction is disposed for each PE in a longitudinal (column) direction, or data transfer routes for connecting PEs in the transverse (row) direction are disposed, so as to effect communication between PEs of different columns. The PE in the longitudinal (column) direction is used for 1-word storage and 1-word operation, for example, and parallel operation is effected for each PE of each column. The present invention provides such a parallel data processor and parallel data processing method.

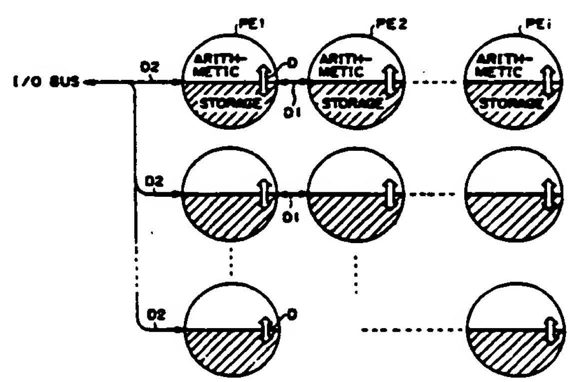


Fig. 4

O : 1-BIT DATA
O1 : INTERNAL DATA
O2 : EXTERNAL DATA

EP 0 539 595 A1

1/0DR 08

Technical Field

The present invention relates to a data processing system and, more specifically, to a data processor for and a data processing method of performing parallel data processing at high speed.

Background Art

Data processors adapted for parallel processing have been developed recently in order to speed up data processing. Fig. 1 illustrates a prior art of such data processors. In Fig. 1A, a data processor 5 is constructed from an instruction fetch unit 1 for receiving an instruction externally input over a bus 6, an instruction decode unit 2 for decoding the received instruction, an operation execute unit 3 for reading operands out of a register file 4 and then performing arithmetic based on the received instruction, and the register file 4 for storing the result of the execution.

The processes of receiving the instruction from the bus 6, decoding it, performing arithmetic, and storing the result of the arithmetic in the register file 4 in Fig. 1A can be indicated by a four-stage pipeline operation as shown in Fig. 1B. That is, in the first prior art, the pipeline processing consists of four stages: an instruction fetch stage; an instruction decode and operand fetch stage; an instruction execute stage; and a result store stage. In the case of this system, therefore, parallel processing can be performed by providing the processor with a plurality of arithmetic perform units and a register file having multiple ports. However, since the number of storage locations in the register file 4 is larger than the number of the arithmetic units 3, it takes long to control which of the storage locations is to be accessed by an arithmetic unit 3. In addition, the number of bus lines which connect the arithmetic units 3 and the register file 4 becomes very large. For example, if three arithmetic units for 32-bit arithmetic operations are provided, then as many as 32 x 2 (two sets are needed for read and write) x 3 = 192 bus lines will be required however simply it is considered. Further, the routing of the bus lines becomes complicated, making an integrated circuit version difficult. If, on the other hand, the plural arithmetic units 3 and the register file 4 are connected by a common bus, then a large amount of data will flow through the common bus, so that von Neumann bottleneck occurs. Thus, there is a problem in that the instruction execute stage and the result store stage become slowed.

Fig. 2 illustrates a second prior art of data processors, which uses a logic-in memory system in which the arithmetic facility and the storage facility are integrated on the same chip and performs serial-by-bit arithmetic. The chip is composed of, say, 64K (2^{16}) basic gate cells each comprising a 4K (2^{12})-bit external memory 7, a serial arithmetic and logic unit (ALU) 8, and an internal flag register 9. All the buses are 1 bit in width.

In Fig. 2, two pieces of data A and B stored in separate locations in the external memory 7 are read out as input data from the external memory 7 to the ALU 8, arithmetic is performed by the ALU 8, and the result is stored in the external memory 7 again. The flag register 9 generates a condition code for arithmetic to be performed by the ALU 8 and is used, for example, to store an overflow bit and a carry output, and re-enter a carry-in to the high-order bit into the ALU 8 when the result of arithmetic by the ALU 8 causes overflow.

In performing 32-bit arithmetic processing, by way of example, the prior art of Fig. 2 requires that a process of reading data to be operated on from the external memory 7 be performed 32 times and a process of writing the result into the memory be performed 32 times. Thus, a problem arises in that communication time between the external memory 7 and the ALU 8 becomes long, making speeding up of data processing impossible.

Disclosure of Invention

It is an object of the present invention to provide a processor element which performs high-speed data communication with another processor element while circumventing von Neumann bottleneck and performs parallel data processing using the result of the communication, and an architecture of a data processing system using such processor elements.

It is the other object of the present invention to provide a parallel data processor suitable for a semiconductor integrated circuit version.

Fig. 1 is a block diagram illustrating fundamentals of a processor element as a data processor of the invention. As shown, a processor is equipped with a 1-bit arithmetic means 11 for performing arithmetic on 1-bit data to be operated on, e.g., a 1-bit arithmetic unit, storage means 12 for storing the data to be operated on and the result of the arithmetic by the 1-bit arithmetic means 11, e.g., a 1-bit memory unit, and communication means 13 for permitting communication between each of the 1-bit arithmetic means 11 and the storage means 12 and another processor element. In Fig. 3, the storage means 12 is incorporated into the processor element to construct a logic-in memory. Since the output of the 1-bit arithmetic means 11 is connected to the storage means 12, consisting of, for example, 1-bit memory, directly with no intervening bus, it be-

comes unnecessary to spend selecting a location to be written in within the storage means 12. Therefore, when the processor element performs pipeline operation, the result store (write) stage can be shortened considerably, improving arithmetic speed of the processor element.

Fig. 4 is a diagram for use in explanation of a concept of data processing in a parallel data processing system in which processor elements, each configured as shown in Fig. 3, are arranged in a matrix. In the figure, suppose that 32 processor elements are arranged in each of the top-to-bottom and left-to-right directions and thus the total number of processor elements is 1024. Suppose that the processor elements arranged in the left-to-right direction are connected by, for example, a common data bus not shown, while the processor elements arranged in the top-to-bottom direction are connected by, for example, a data transfer line for transferring a carry, etc. Further, suppose that each of the processor element is connected to an I/O bus for inputting/outputting data from the outside of the system.

In Fig. 4, the processors in the top-to-bottom direction store, for example, 1-word data and perform arithmetic on that data. For example, each of the elements in a column stores a corresponding bit of 32-bit data, with the least significant bit stored in the lowest element in the column, and performs arithmetic on the stored bit.

In Fig. 4, from each of the processor elements PE1 in the leftmost column to a corresponding one of the 32 processor elements PE 2 in the second column from the left is transferred 1-bit internal data D1 stored in each processor using the common data bus not shown or inter-processor-element data transfer, or data D2 which is externally input over the I/O bus. Each of the elements PE2 performs arithmetic on the transferred data D1 or D2 and 1-bit data D stored in it. The result is stored in the storage means 12 in each of the elements PE2. Alternatively, the result of the arithmetic may be output to outside over the I/O bus as required.

In Fig. 4, when complicated processing requiring a pipeline process to be performed a large number of times is performed, the pipeline process can be speeded up by repeating, a required number of times, combined processes of fetching an externally input instruction, decoding it, transferring necessary data to each processor according to the result of the instruction decoding, performing arithmetic, and storing the result. Further, since processor elements each comprising 1-bit arithmetic means and storage means are arranged in the form of an array, the degree of parallelism can be much improved by performing parallel arithmetic for each word.

Since, as shown in Fig. 4, the arrangement of the processor elements and wiring among the processor elements is made regular, the data processor according to the present invention is suitable for an integrated circuit version.

Brief Description of Drawings

Fig. 1A, B is a diagram explanatory of a first prior-art data processor;

Fig. 2 is a diagram explanatory of a second prior-art data processor;

Fig. 3 is a block diagram illustrating fundamentals of a processor element as a data processor of the present invention;

Fig. 4 is a diagram illustrating the concept of data processing in a parallel data processing system using the processor element of the present invention;

Fig. 5 is a block diagram of a first embodiment of the parallel data processing system;

Fig. 6 is a detailed block diagram of the processor element of Fig. 5;

Fig. 7 is a block diagram of an embodiment of a PE controller;

Fig. 8 is a block diagram of a second embodiment of the parallel data processing system;

Fig. 9 is a detailed block diagram of the processor element of Fig. 8;

Fig. 10 is a flowchart of an embodiment of the parallel data processing of the present invention;

Fig. 11 is a diagram for a supplemental explanation of the flowchart of Fig. 10;

Figs. 12A, 12B and 12C are diagrams for use in explanation of a pipeline process of the present invention; and

Fig. 13 is a diagram of a concrete example of arithmetic processing in an embodiment of the present invention.

Best Mode of Carrying Out the Invention

Fig. 5 is a block diagram of a first embodiment of a parallel data processing system according to the present invention. In the figure, processor elements 10 are arranged in the top-to-bottom and left-to-right directions to form a matrix. The top-to-bottom and left-to-right directions correspond to one word. For example, when one word is 32 bits, 32 processor elements 10 are arranged in a column. The processor elements 10 in each column permit parallel arithmetic. In the left-to-right direction a common data bus 17 is provided for processor elements (PE) in each row. In the top-to-bottom direction the PEs 10 are connected to each other by a data transfer line 18 for transferring a carry, etc., as will be described later. In common with the parallel processor elements

required to control storage locations in the register file, as required in the prior art, the common bus is not used on a time-sharing basis in transferring the results of arithmetic from the arithmetic units to the register file, and data transfers are not delayed. In addition, the PEs themselves are capable of not only high-speed operation but also parallel operation when arranged in an array, thus much increasing the speed of processing.

Fig. 11 is a diagram for use in supplemental explanation of the flowchart of Fig. 10. In the figure, each controller 30 sends various control signals to processor elements in the top-to-bottom direction, that is, to a processor element array 20, in accordance with the result of decoding of an instruction by the instruction decode unit 16. Each 1-bit arithmetic unit 21 comprising the processor element array 20 performs arithmetic using input data DIN to be operated on which is externally input over the I/O bus 19. The result is output to outside via the 1-bit memory unit 22 as arithmetic result data DOUT.

Figs. 12A, 12B and 12C are diagrams for use in explanation of stages of pipeline processing according to the present invention which is performed in the processor elements PE1, PE2 and PE3 arranged in the direction of a row. In the figure, data transfer (T) is pipeline-processed subsequent to the two stages of instruction fetching (F) and decoding (D). Control signals from the instruction decode unit are applied to the processor elements PE1, PE2 and PE3 in parallel, so that the execution of arithmetic (E) and the writing of the result (W) are processed in parallel in the processor elements. Since the data transfer (T) time after the execution (E) can almost be neglected, the writing (W) of the result can be performed very fast.

Next, in the present invention, the processor elements arranged in a two-dimensional matrix with n rows and m columns can substantially be split into groups of processor elements for respective independent arithmetic operations.

Fig. 13 is a diagram for use in explanation of an embodiment of such split usage of a parallel data processing system, which illustrates provision of more than one common data bus 55 for processor elements arranged from left to right. That is, this embodiment can be considered as including more than one common data bus in the embodiment of the first parallel data processing system (Fig. 5) in which one common data bus 17 is provided for processor elements in the left-to-right direction.

In Fig. 13, there are shown, for simplicity, three processor elements PE1, PE2 and PE3 in the left-to-right direction and only one processor in the top-to-bottom direction. However, 32 elements exist both in the left-to-right direction and in the top-to-bottom direction. The PEs in the top-to-bottom direction constitute one word. Each element (PE) is shown, for simplicity, as comprising an arithmetic unit 51 corresponding to the 1-bit arithmetic means of Fig. 3, a storage unit 52 corresponding to the storage means, and registers 53 and 54 which temporarily store operand data transferred from other PEs over the common data buses 55 and whose stored contents are read out by high-speed clocks at arithmetic time. In this example, the communication units in Fig. 6 are omitted. In addition, although a plurality of common data buses are shown above (N) and below (S) the PEs, as the common data buses 55, this is not restrictive.

It is assumed that the system is split for use, and the common data buses 55 are larger in number than is necessary to an instruction to perform simultaneous parallel arithmetic.

In Fig. 13, for example, data "A" is transferred from the storage section 52 in PE1 on the left (W) to the register 53 in PE2 over the upper common data bus 55. And data "B" is transferred from the storage section 52 in the PE3 on the right (E) to the register 54 in the PE2 over the lower common data bus 55. For example, arithmetic "A + B" is performed by the arithmetic unit 51 in the PE2 and then the result is stored in the storage unit 52 in the PE2.

At the same time as such operations are performed by the PE1 to PE3, PE4 to PE6 (assumed to be to the right of the PE3), not shown, can execute addition of two pieces of data "C" and "D" in exactly the same manner. In this case, out of the common data buses 55, buses that are not used by the PE1 to PE3 are used by the PE4 to PE6, which substantially splits the parallel data processing system for use and ensures very efficient system usage.

When the processor elements, as shown in Fig. 9, are arranged in an array as shown in Fig. 8, the processor elements can be split in word units and operated in parallel in the same manner as described above by controlling the output communication units for the E and W directions.

According to the present invention, naturally the processor element shown in Fig. 6 which uses common data buses and the processor element shown in Fig. 9 which uses an inter-processor element transfer path in the direction of row may be combined into one processor element, and the resulting processor elements may be arranged in an array.

Industrial Applicability

According to the present invention, the result writing (W) stage can be speeded up in each PE,

and the degree of parallelism can be improved by arranging PEs in an array to much increase the speed of data arithmetic operation. Moreover, the device of the present invention is suitable for a one-chip integrated circuit version because common data buses and inter-PE transfer paths are able to be wired regularly in the left-to-right and the top-to-bottom direction and PEs can be formed in the same pattern. Furthermore, since the need for a large number of bus lines to be driven to transfer data as in a conventional parallel data processor is eliminated, the processing speed also improves.

Accordingly, the data processors of the present invention can be used in various types of parallel data processing systems such as an image processing system, etc.

Claims

1. A processor element characterized by comprising:
   arithmetic means (11) for performing arithmetic on data to be operated on;
   storage means (12) for storing the data to be operated on or the result of the arithmetic;
   communication means (13) for permitting communication between said arithmetic means (11) and another processor element; and
   access means for accessing said storage means (12) independently of said communication means.

2. A processor element according to claim 1, characterized in that said arithmetic means is one bit.

3. A processor element according to claim 1, characterized in that said storage means is one bit.

4. A processor element according to claim 1, characterized in that said storage means stores data of one or more bits to be operated on and/or the result of the arithmetic.

5. A processor element according to claim 1, characterized in that said storage means (12) stores data to be operated on as a result of a communication process of said communication means (13) or data to be operated on which is externally input without being routed through said communication means (13).

6. A processor element according to claim 1, characterized in that said communication means is connected to a data transfer line for transferring a carry and to a common bus.

7. A processor element according to claim 1, characterized in that said communication means is connected to a data transfer line for transferring a carry transfer line and a data transfer line to an adjacent processor element.

8. A processor element comprising:
   an arithmetic unit;
   a memory unit connected to said arithmetic unit;
   a communication unit for selectively bypassing communication between said arithmetic unit and another processor element and data; and
   an interface for accessing said memory unit independently of said communication unit.

9. A data processor having processor elements arranged in a matrix characterized in that
   each of said processor elements comprises:
   arithmetic means (11) for performing arithmetic on data to be operated on;
   storage means (12) for storing the data to be operated on or the result of the arithmetic;
   communication means (13) for permitting communication between said arithmetic means (11) and another processor element; and
   access means for accessing said storage means (12) independently of said communication means,
   a signal generated by each of said processor elements being transferred in the direction along a column and common buses being provided in the direction along a row to thereby execute arithmetic operation in parallel.

10. A data processor according to claim 9, characterized by providing at least as many common data buses as there are simultaneous parallel arithmetic operations along first, second and third processor elements arranged in a row, and performing an arithmetic operation in parallel which performs arithmetic on the contents of said storage unit of said first processor element and the contents of said storage unit of said second processor element in said arithmetic unit of said third processor element and stores the result of the arithmetic in said storage unit of said third processor element.

11. A data processor according to claim 9, characterized in that said processor elements arranged in a matrix are assembled into one chip.

12. A data processor according to claim 9, characterized in that said parallel arithmetic operation is performed by causing each column of processor elements to perform arithmetic corresponding to one instruction fetched in parallel.

13. A data processor according to 9, characterized in that said parallel arithmetic operation is performed on a pipeline basis.

14. A data processor according to claim 9, characterized in that one controller is provided for each group of processor elements arranged in a column, and said controller comprises a common bus controller for determining whether the result of arithmetic is to be output onto a common bus line, an arithmetic controller for controlling what arithmetic is to be performed by said arithmetic unit, and a memory I/O controller for performing control between a memory and an I/O bus.

15. A data processor having processor elements arranged in a matrix characterized in that

each of said processor elements comprises:

arithmetic means (11) for performing arithmetic on data to be operated on;

storage means (12) for storing the data to be operated on or the result of the arithmetic;

communication means (13) for permitting communication between said arithmetic means (11) and another processor element; and

access means for accessing said storage means (12) independently of said communication means,

a signal generated by each of said processor elements being transferred in the direction of a column and data being transferred in the direction of a row to thereby execute arithmetic operation in parallel.

16. A data processor according to claim 15, characterized by providing at least as many common data buses as there are simultaneous parallel arithmetic operations along first, second and third processor elements arranged in a row, and executing an arithmetic operation in parallel which performs arithmetic on the contents of said storage unit of said first processor element and the contents of said storage unit of said second processor element in said arithmetic unit of said third processor element and stores the result of the arithmetic in said storage unit of said third processor element.

17. A parallel data processing system characterized by comprising:

n x m unit data processor elements comprising m sets of n unit data processing elements connected by a data transfer path,

n common data buses (17) each for m corresponding unit data processor elements in m sets of n unit data processor elements;

an instruction fetch unit (15) for fetching an instruction;

an instruction decode unit (16) for decoding the instruction fetched by said instruction fetch unit (15); and

m processor element controllers (30) for outputting a control signal for data processing to each of n unit data processor elements in each of said m sets according to the instruction decoded by said instruction decode unit (16).

18. A parallel data processing system according to claim 17, characterized in that each of said data processor elements comprises:

a 1-bit arithmetic unit (21) for performing arithmetic on 1-bit data to be operated on;

a 1-bit memory unit (22) for storing an output of said 1-bit arithmetic unit (21) or data input from the outside of the system and outputting the stored content to said 1-bit arithmetic unit (21);

a read/write interface (24) for controlling data input/output between said 1-bit memory unit (22) and the outside of the system;

an input communication unit (23) for inputting data other than a carry input which is input from said common data buses (17) or an adjacent unit data processor element of said n unit data processor elements to said 1-bit arithmetic unit (21) and inputting a carry input from said adjacent data processor element to a carry input terminal (23) of said 1-bit arithmetic unit (21);

an output communication unit (25) for selectively outputting a carry output of said 1-bit arithmetic unit (21) or an output of said 1-bit memory unit (22) to said adjacent data processing element; and

a tri-state buffer (27) for outputting an output of said 1-bit memory unit (22) to said common data buses (17).

19. A parallel data processing system according to claim 17, characterized in that each of said m processor element controllers (30) for each of m sets of n data processor elements comprises:

a memory controller (33) for outputting a data input/output control signal to said

read/write interface (24);

an arithmetic controller (32) for outputting an arithmetic control signal to said 1-bit arithmetic unit (21) and said input communication unit (23); and

a data transfer controller (31) for outputting a data transfer control signal to said input communication unit (23), an output communication unit (25) and said tri-state buffer (27).

20. A parallel data processing system according to claim 17, characterized in that each of said m sets of unit data processing elements are in charge of processing data in word units, and processes (S42 to S44) of, according to an instruction fetched by said instruction fetch unit (15) and decoded by said instruction decode unit (16), performing necessary data transfers between unit bit-corresponding data processing units in said m sets of unit data processing units (S42), performing arithmetic by each unit data processing unit (S43), and storing the result of arithmetic in storage means (S44) are performed a required number of times according to the contents of the instruction decoded.

21. A parallel data processing system according to claim 17, characterized in that, in place of one common data bus for each set of m corresponding unit data processor elements in said m sets of n unit data processor elements, at least a plurality of (l) common data buses required at a time of execution of simultaneous parallel arithmetic instruction are provided, and the total number of n x l common data buses (55) are provided.

22. A parallel data processing system comprising:

n x m unit data processor elements arranged in a two-dimensional matrix with n columns and m rows, said elements being connected by first m data transfer lines in the direction along a column and by second n data transfer lines in the direction along a row;

an instruction fetch unit (15) for fetching an instruction;

an instruction decode unit (16) for decoding the instruction fetched by said instruction fetch unit (15); and

m processor element controllers (30) each for outputting a control signal for data processing to n unit data processing units in a corresponding one of m columns.

23. A parallel data processing system according to claim 22, characterized in that each of said data processor elements comprises:

a 1-bit arithmetic unit (21) for performing arithmetic on 1-bit data to be operated on;

a 1-bit memory unit (22) for storing an output of said 1-bit arithmetic unit (21) or data input from the outside of the system and outputting the stored content to said 1-bit arithmetic unit (21);

a read/write interface (24) for controlling data input/output between said 1-bit memory unit (22) and the outside of the system;

an input communication unit (23) for inputting data other than a carry input which is input from an adjacent unit data processor element of n unit data processor elements in the direction along a column or m unit data processor elements in the direction along a row to said 1-bit arithmetic unit (21) and inputting a carry input from said adjacent data processor element to a carry input terminal (23) of said 1-bit arithmetic unit (21);

an output communication unit (25) for selectively outputting a carry output of said 1-bit arithmetic unit (21) or an output of said 1-bit memory unit (22) to said adjacent data processing element in the direction along a column; and

second and third output communication units (35, 36) for selectively outputting data input from an adjacent unit data processor element of m data processor elements in a row or an output of said 1-bit memory unit (22) to said adjacent unit data processor elements in the direction opposite to said adjacent unit data processor element of m data processor elements in a row.

24. A parallel data processing system according to claim 22, characterized in that each of m PE controllers (30) for m sets of n unit data processor elements comprises a memory controller (33) for outputting a data input/output control signal to said read/write interface (24), an arithmetic controller (32) for outputting an arithmetic control signal to said 1-bit arithmetic unit (21) and said input communication unit (23), and a data transfer controller (31) for outputting a data transfer control signal to said input communication unit (23) and said first, second and third output communication units (25, 35, 36).

25. A parallel data processing system according to claim 22, characterized in that each of said m sets of n unit data processing elements are in charge of processing data in word units, and processes (S42 to S44) of, according to an instruction fetched by said instruction fetch unit

(15) and decoded by said instruction decode unit (16), performing necessary data transfers between bit-corresponding data processing units in said m sets of unit data processing units (S42), performing arithmetic by each unit data processing unit (S43), and storing the result of arithmetic in storage means (S44) are performed a required number of times according to the contents of the instruction decoded.

26. A data processing method characterized by comprising the steps of:
performing arithmetic processing on data bit by bit on the basis of an instruction;
writing the result of the arithmetic into a 1-bit memory; and
transferring data read out of said 1-bit memory to outside, thereby performing unit data processing.

27. A data processing method according to claim 26, characterized in that said unit data processing in each of processor elements arranged in an array, processing of one word is performed by processor elements in a column, and two or more words are processed in parallel.

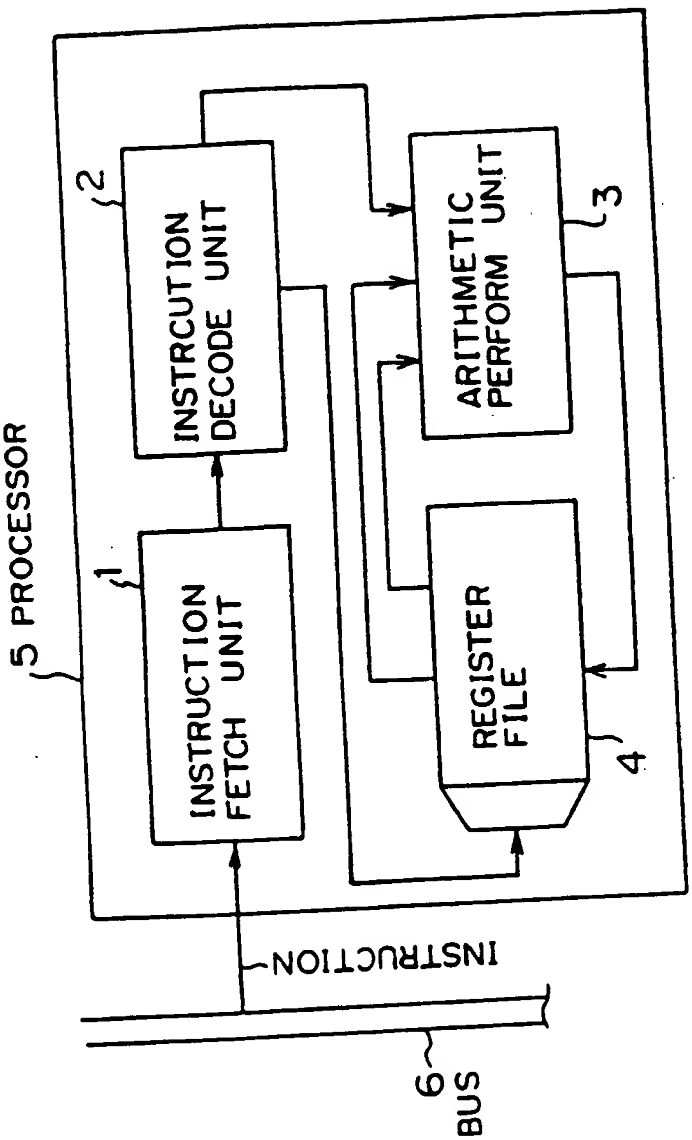

5 PROCESSOR
1
INSTRUCTION FETCH UNIT
2
INSTRCUTION DECODE UNIT
ARITHMETIC PERFORM UNIT
3
REGISTER FILE
4
INSTRUCTION
6
BUS

Fig. 1A

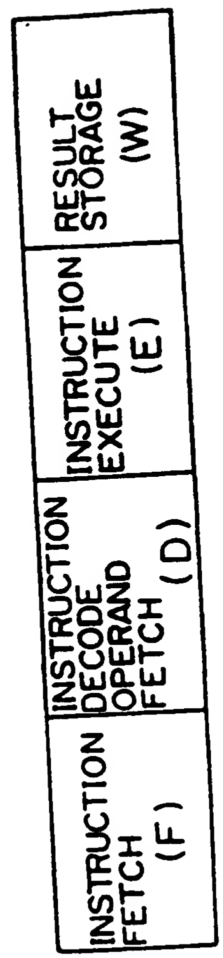

INSTRUCTION FETCH (F)
INSTRUCTION DECODE OPERAND FETCH (D)
INSTRUCTION EXECUTE (E)
RESULT STORAGE (W)

Fig. 1B

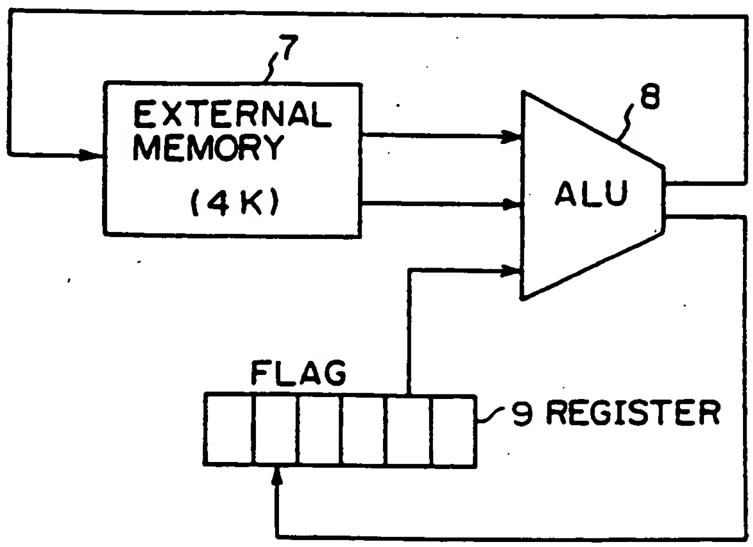


Fig. 2

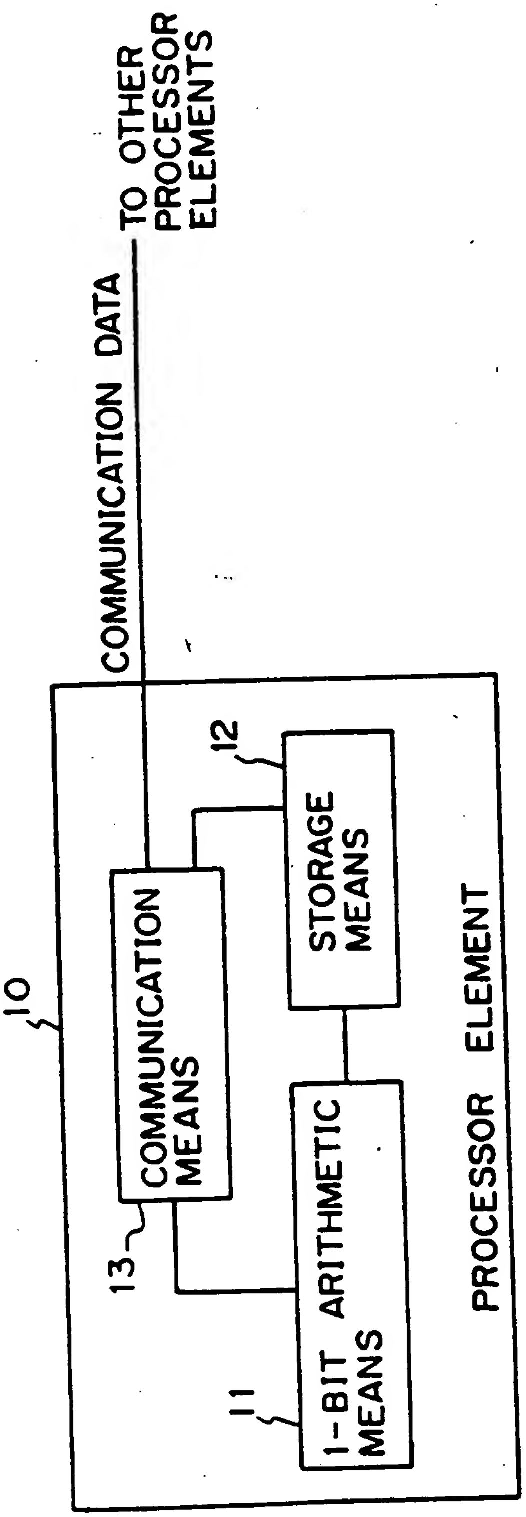
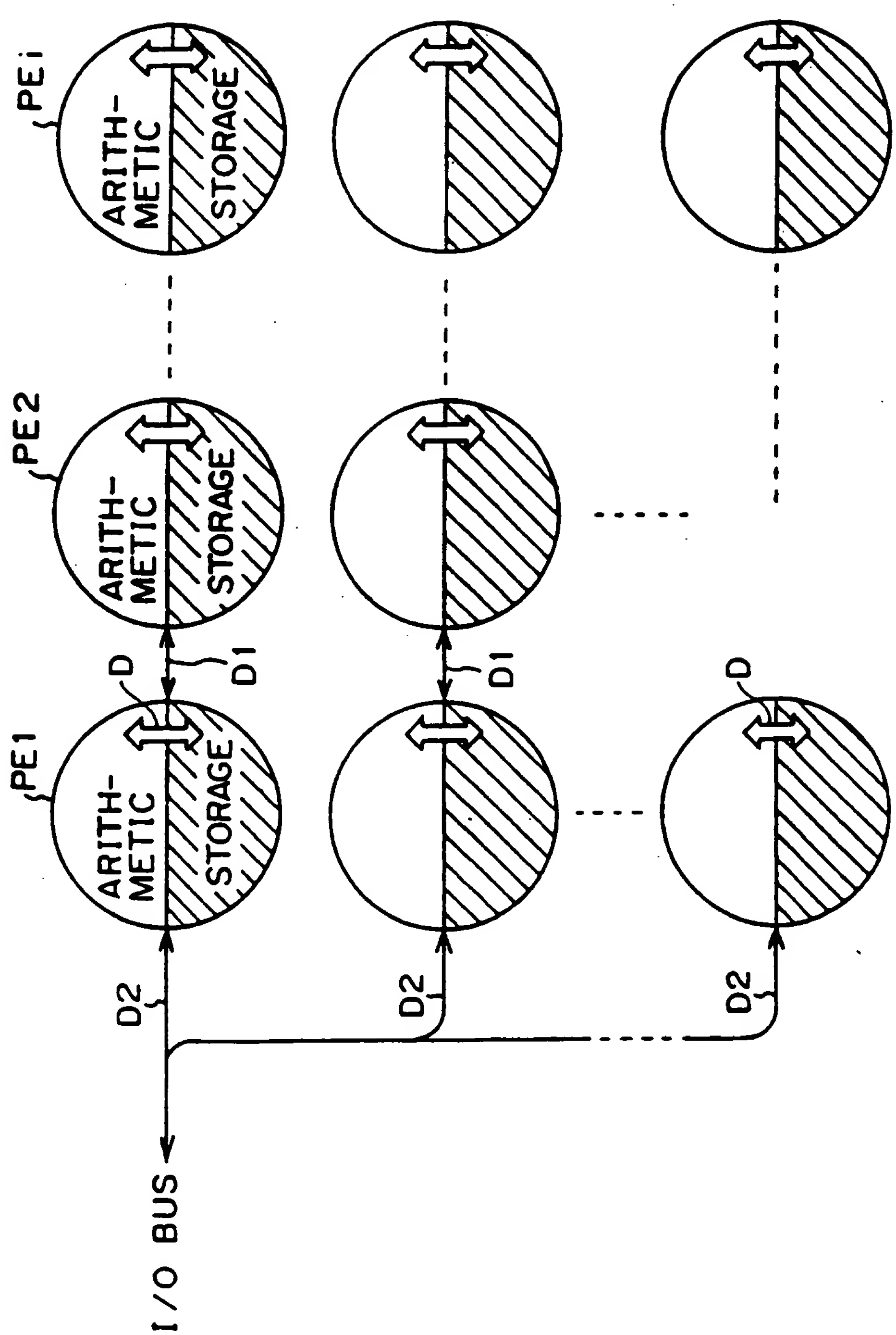

TO OTHER PROCESSOR ELEMENTS
COMMUNICATION DATA
10
12
STORAGE MEANS
COMMUNICATION MEANS
13
11
1-BIT ARITHMETIC MEANS
PROCESSOR ELEMENT

Fig. 3



D : 1-BIT DATA
D1 : INTERNAL DATA
D2 : EXTERNAL DATA

Fig. 4

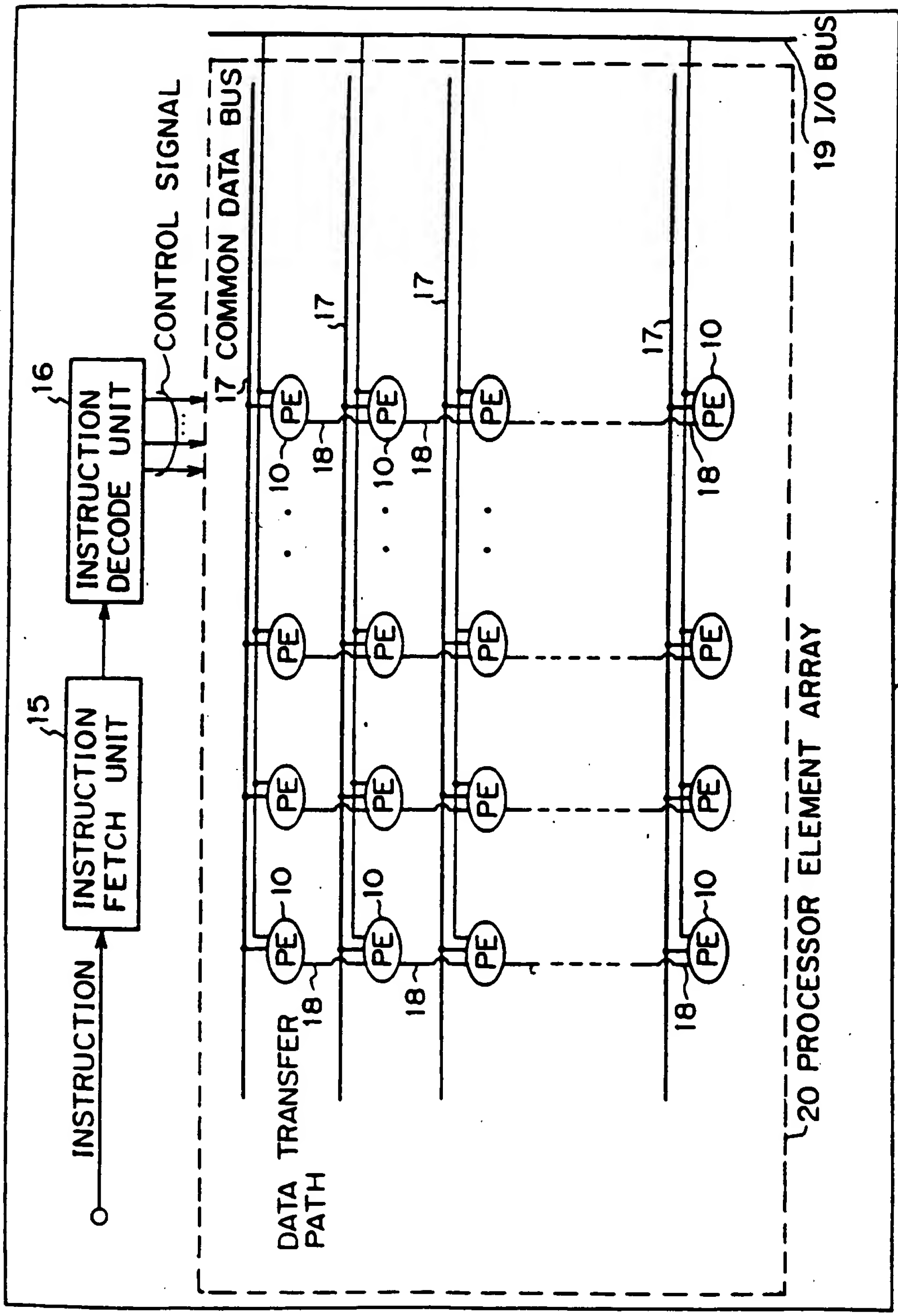

INSTRUCTION
15
INSTRUCTION FETCH UNIT
16
INSTRUCTION DECODE UNIT
CONTROL SIGNAL
17 COMMON DATA BUS
19 I/O BUS
17
10
18
PE
DATA TRANSFER PATH
20 PROCESSOR ELEMENT ARRAY
50 SEMICONDUCTUR SUBSTRATE

Fig. 5

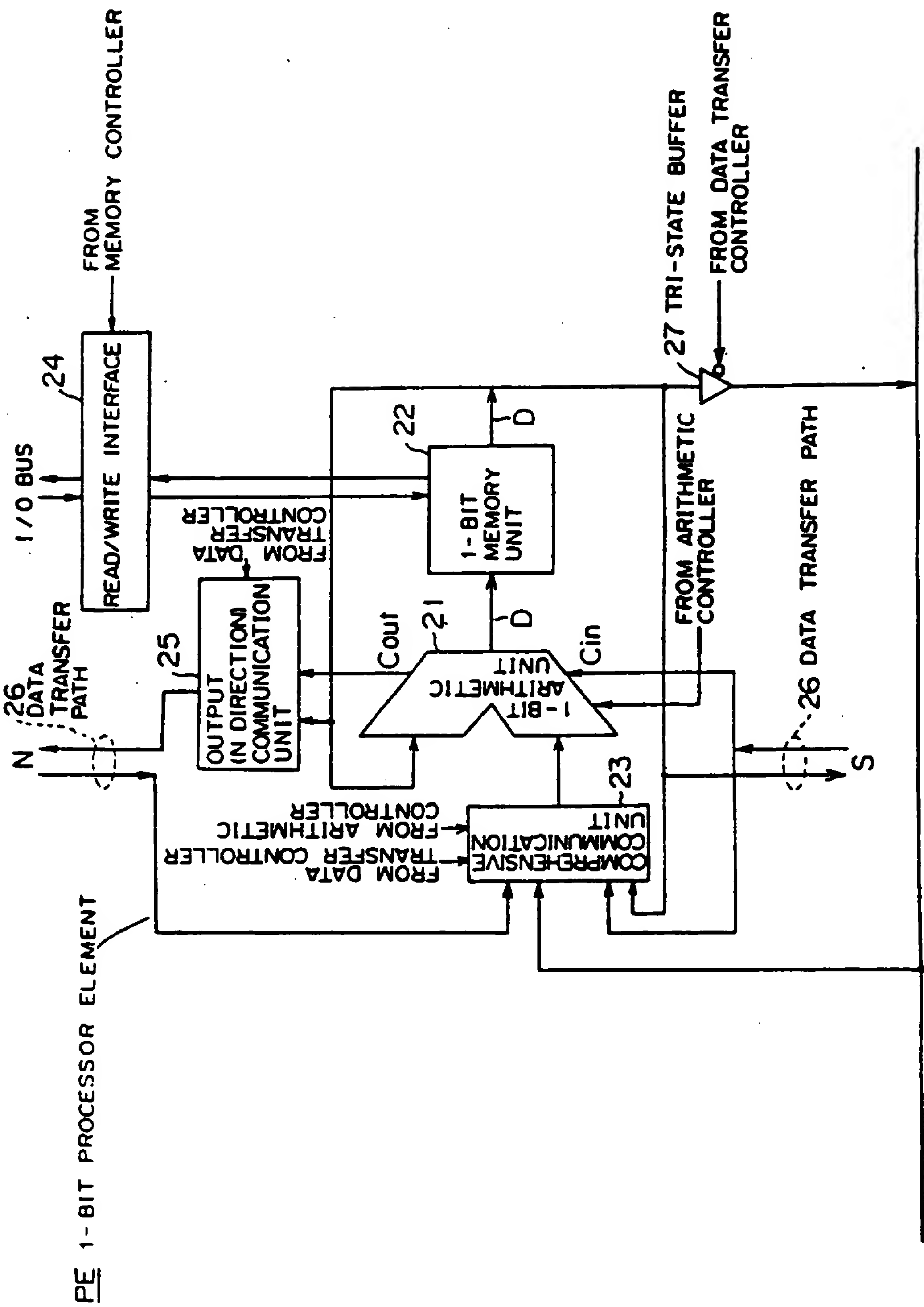

FROM MEMORY CONTROLLER
24
READ/WRITE INTERFACE
I/O BUS
22
1-BIT MEMORY UNIT
D
FROM DATA TRANSFER CONTROLLER
OUTPUT (IN DIRECTION) COMMUNICATION UNIT
25
26 DATA TRANSFER PATH
N
Cout
21
D
Cin
1-BIT ARITHMETIC UNIT
23
COMPREHENSIVE COMMUNICATION UNIT
FROM DATA TRANSFER CONTROLLER
FROM ARITHMETIC CONTROLLER
27 TRI-STATE BUFFER
FROM DATA TRANSFER CONTROLLER
FROM ARITHMETIC CONTROLLER
26 DATA TRANSFER PATH
S
PE 1-BIT PROCESSOR ELEMENT
17 COMMON BUS

Fig. 6

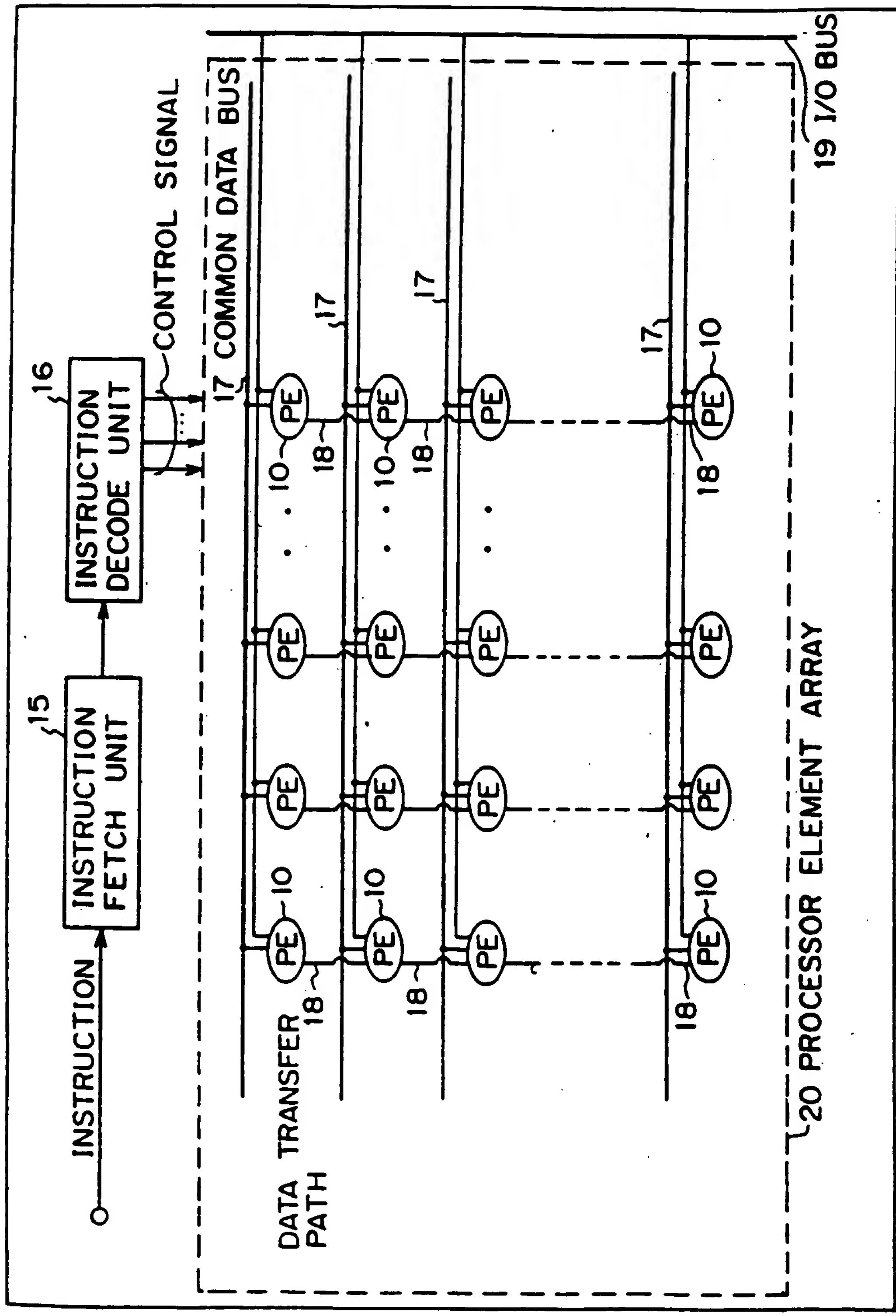

INSTRUCTION
15
INSTRUCTION FETCH UNIT
16
INSTRUCTION DECODE UNIT
CONTROL SIGNAL
17 COMMON DATA BUS
17
19 I/O BUS
PE
10
18
DATA TRANSFER PATH
20 PROCESSOR ELEMENT ARRAY
50 SEMICONDUCTUR SUBSTRATE

Fig. 5

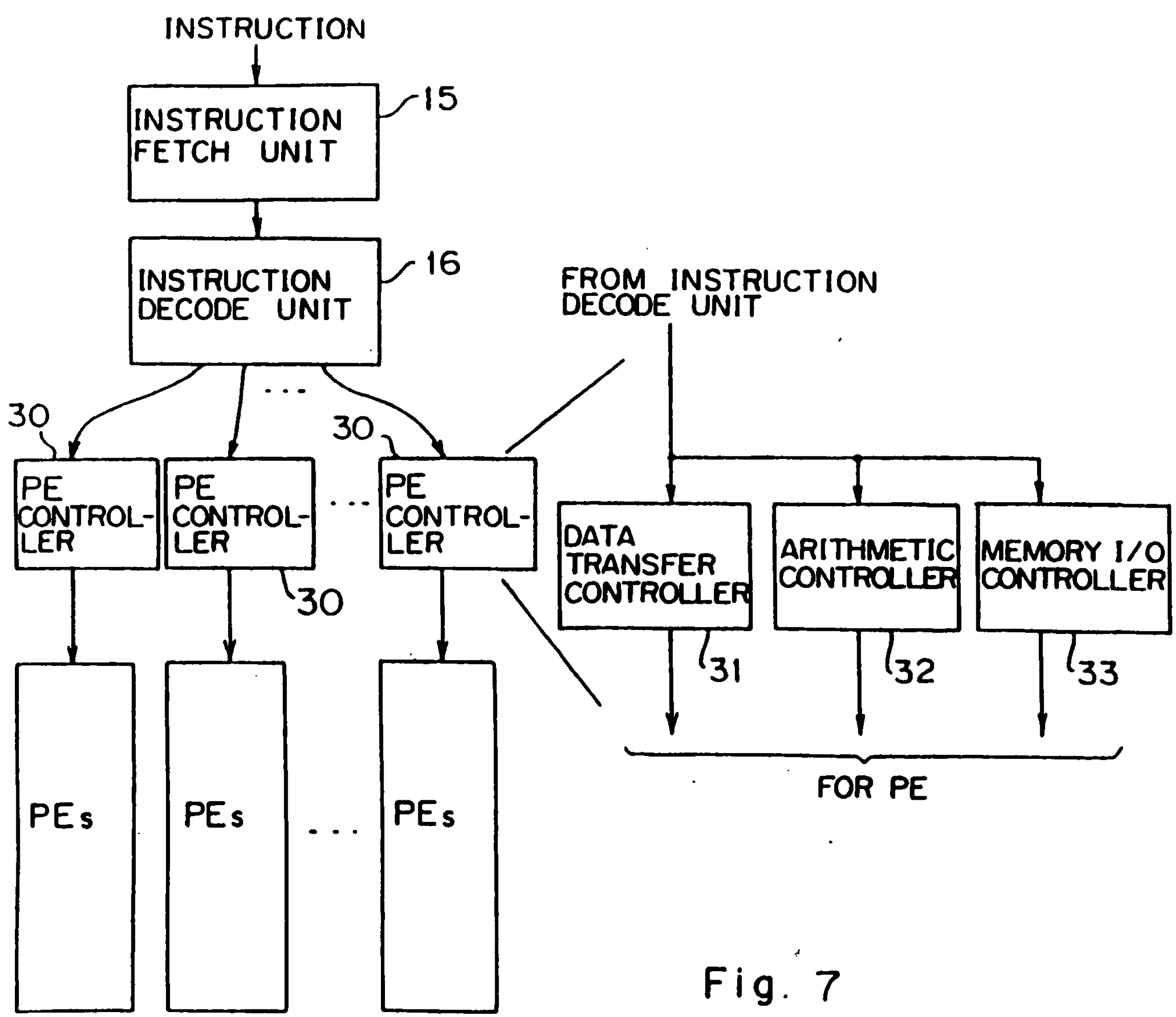

INSTRUCTION
INSTRUCTION FETCH UNIT
15
INSTRUCTION DECODE UNIT
16
30
PE CONTROL-LER
PE CONTROL-LER
PE CONTROL-LER
30
30
PEs
PEs
PEs
FROM INSTRUCTION DECODE UNIT
DATA TRANSFER CONTROLLER
ARITHMETIC CONTROLLER
MEMORY I/O CONTROLLER
31
32
33
FOR PE

Fig. 7

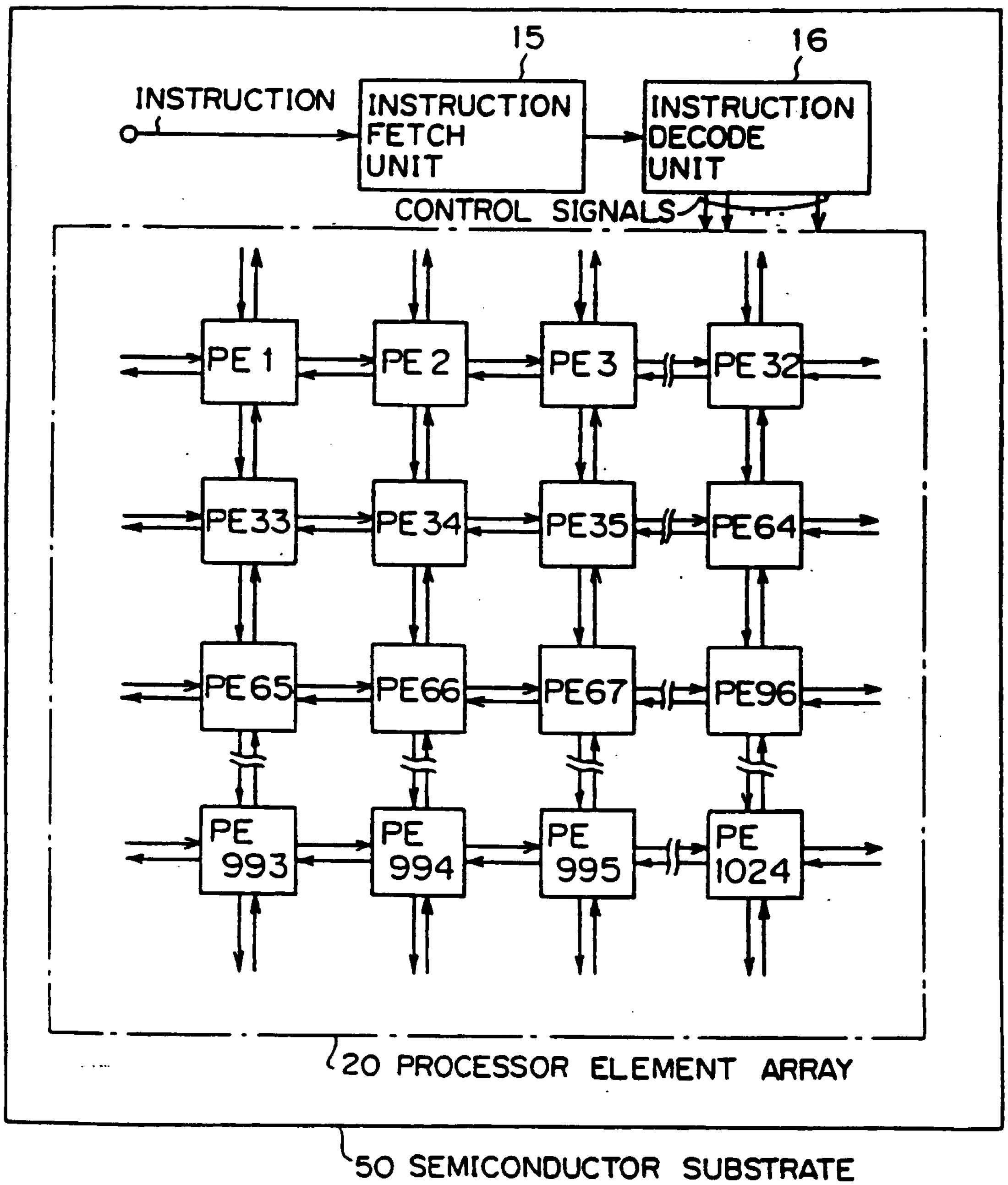

15
16
INSTRUCTION
INSTRUCTION FETCH UNIT
INSTRUCTION DECODE UNIT
CONTROL SIGNALS
PE 1
PE 2
PE 3
PE 32
PE33
PE 34
PE35
PE64
PE 65
PE66
PE67
PE96
PE 993
PE 994
PE 995
PE 1024
20 PROCESSOR ELEMENT ARRAY
50 SEMICONDUCTOR SUBSTRATE

Fig. 8

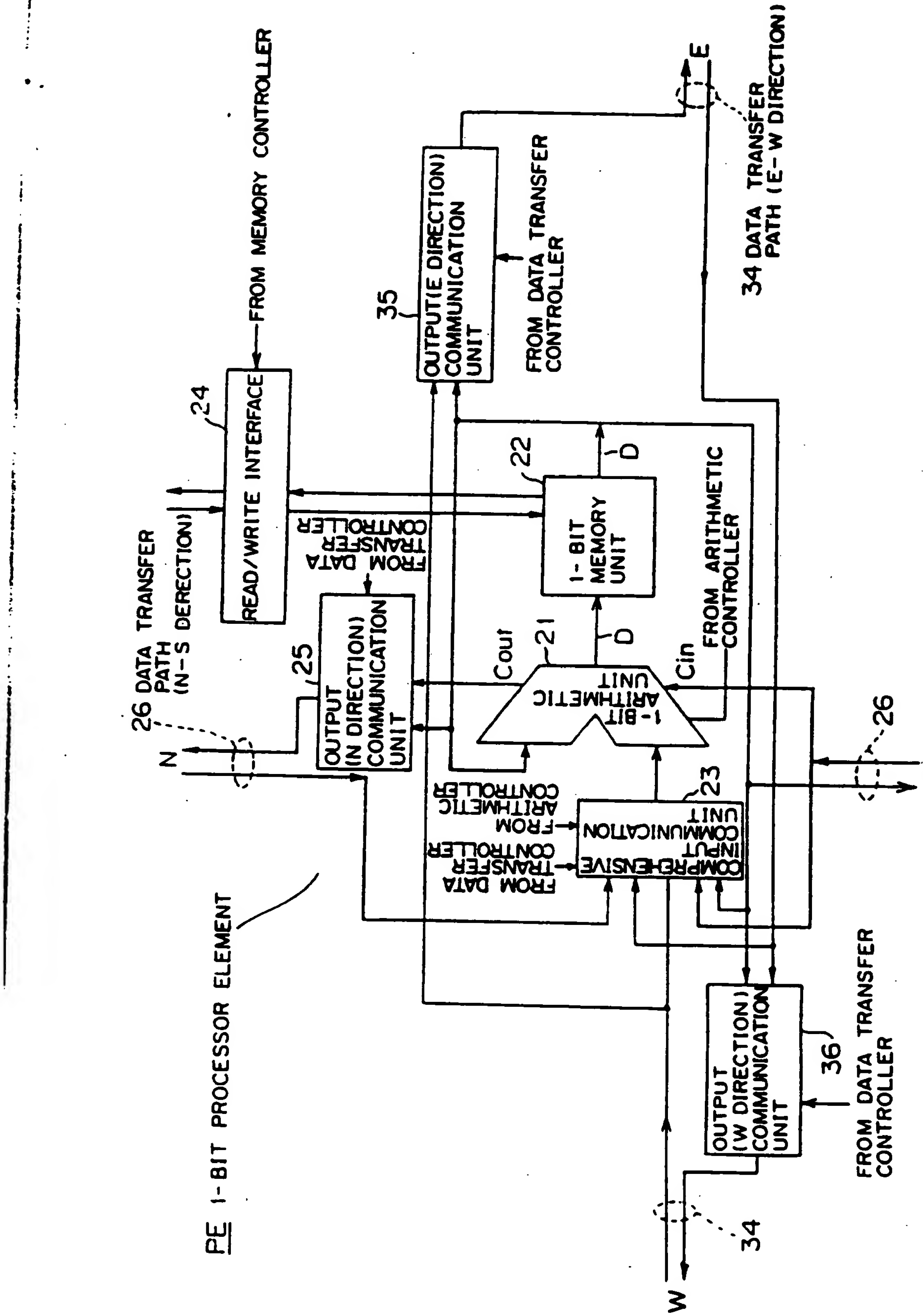

PE 1-BIT PROCESSOR ELEMENT
FROM MEMORY CONTROLLER
24
READ/WRITE INTERFACE
26 DATA TRANSFER PATH (N-S DERECTION)
FROM DATA TRANSFER CONTROLLER
25
OUTPUT (N DIRECTION) COMMUNICATION UNIT
N
35
OUTPUT(E DIRECTION) COMMUNICATION UNIT
FROM DATA TRANSFER CONTROLLER
E
34 DATA TRANSFER PATH (E-W DIRECTION)
22
1-BIT MEMORY UNIT
D
21
Cout
1-BIT ARITHMETIC UNIT
Cin
FROM ARITHMETIC CONTROLLER
23
COMPREHENSIVE INPUT COMMUNICATION UNIT
FROM DATA TRANSFER CONTROLLER
FROM ARITHMETIC CONTROLLER
26
S
36
OUTPUT (W DIRECTION) COMMUNICATION UNIT
FROM DATA TRANSFER CONTROLLER
34
W

Fig. 9

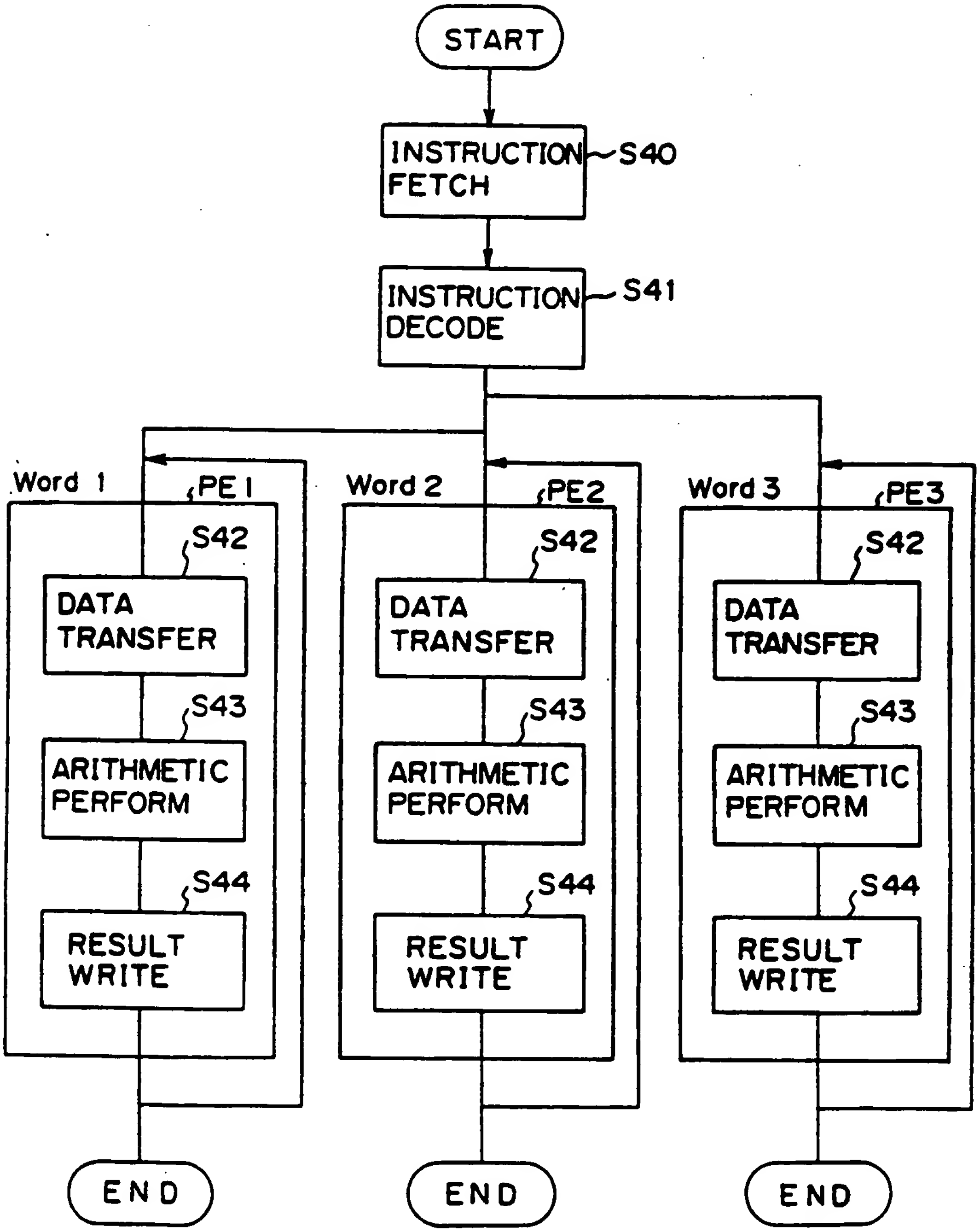

START
INSTRUCTION FETCH
S40
INSTRUCTION DECODE
S41
Word 1
PE1
Word 2
PE2
Word 3
PE3
S42
DATA TRANSFER
S43
ARITHMETIC PERFORM
S44
RESULT WRITE
END

Fig. 10